



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/547,290	04/11/2000	Nir N. Shavit	1004-4664	4986

22120 7590 03/26/2004

ZAGORIN O'BRIEN & GRAHAM, L.L.P.  
7600B N. CAPITAL OF TEXAS HWY.  
SUITE 350  
AUSTIN, TX 78731

EXAMINER

BULLOCK JR, LEWIS ALEXANDER

ART UNIT	PAPER NUMBER
----------	--------------

2126

8

DATE MAILED: 03/26/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

**Office Action Summary**

Application No.

09/547,290

Applicant(s)

SHAVIT ET AL.

Examiner

Lewis A. Bullock, Jr.

Art Unit

2126

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☐ Responsive to communication(s) filed on \_\_\_\_.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-23 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-15 and 17-23 is/are rejected.
- 7) ☒ Claim(s) 16 is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 11 April 2000 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. §§ 119 and 120**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a) ☐ All   b) ☐ Some \* c) ☐ None of:  
1. ☐ Certified copies of the priority documents have been received.  
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_.  
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).  
\* See the attached detailed Office action for a list of the certified copies not received.
- 13) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.  
a) ☐ The translation of the foreign language provisional application has been received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)                      4) ☐ Interview Summary (PTO-413) Paper No(s). \_\_\_\_.
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)                      5) ☐ Notice of Informal Patent Application (PTO-152)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 2,3,5-7.                      6) ☐ Other: \_\_\_\_\_

## **DETAILED ACTION**

### ***Allowable Subject Matter***

1. Claim 16 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.
2. The following is a statement of reasons for the indication of allowable subject matter: Claims 16 detail the list is a doubly-linked list susceptible to concurrent operation of opposing-end variants of the pop operation comprising the steps of executing as part of a pop operation, a DCAS atomic update of a list node and both a deleted node indication and list-end identifier corresponding thereto; wherein the deleted node indication marks the corresponding element for subsequent deletion from the list. Applicant's invention of including the step of encoding a delete node indication in performing an atomic non-blocking pop operation on a node would result in deferring nodes that are to be deleted for concurrent deletion (pg. 3, par. 1009) and allowing access entities to have complete concurrent non-blocking shared access to the shared object (doubly linked list) such that there is no need for locks or a mutual exclusion mechanism (pg. 6, par. 1025). The cited prior art of record details some forms of concurrent non-interfering opposing end accesses to a shared doubly linked list data structure, i.e. concurrent enqueueing on one end while dequeuing from the other, however, the access operations are not concurrent atomic non-blocking opposing end pop operations that when concurrently invoked from both ends encode a delete node indication marking the corresponding element for subsequent deletion from the list.

***Claim Rejections - 35 USC § 102***

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1, 3-5, and 11 are rejected under 35 U.S.C. 102(b) as being anticipated by "Non-Blocking Algorithms for Concurrent Data Structures" by PRAKASH.

As to claim 1, PRAKASH teaches a concurrent shared object representation (FIFO queue) comprising: a computer readable encoding for a sequence of zero or more values (doubly linked list of nodes); and access operations defined for access to each of opposing ends of the sequence (enqueue / dequeue / interior deletion / interior insert operations) (pg. 1 and 5-6), wherein execution of any one of the access operations is non-blocking with respect to any other execution of the access operations throughout a complete range of valid states ("pg. 11, "So, whenever there are two or more elements in the queue, there can be two concurrent streams, an enqueueing stream accessing the tail of the queue, and a dequeueing stream accessing the head of the queue."), including one or more boundary condition states (queue empty) (pg. 22-

24, wherein algorithm considers whether queue is empty for enqueueing), and wherein, at least for those of the valid states other than the one or more boundary condition states (queue empty), opposing-end ones of the access operations are disjoint ("pg. 11, "So, whenever there are two or more elements in the queue, there can be two concurrent streams, an enqueueing stream accessing the tail of the queue, and a dequeuing stream accessing the head of the queue.").

As to claim 3, PRAKASH teaches the encoding includes a linked-list of nodes representing the sequence (doubly linked list) (pg. 5); and wherein the one or more boundary condition states include one or more empty states (pg. 22-24, wherein algorithm considers whether queue is empty for enqueueing).

As to claim 4, PRAKASH teaches the access operations include push (enqueue) and pop (dequeue) operations (pg. 5-6).

As to claim 5, PRAKASH teaches the access operations further include delete operations (interior delete) (pg. 5-6).

As to claim 11, PRAKASH teaches the linked-list of nodes is a doubly-linked list thereof (doubly linked list) (pg. 5).

5. Claims 12-15, 18 and 19 are rejected under 35 U.S.C. 102(b) as being anticipated by "A Non-blocking Algorithm for Shared Queues Using Compare and Swap" by PRAKASH.

As to claim 12, PRAKASH teaches a method of managing access to a dynamically allocated list (singly linked list) susceptible to concurrent operations (enqueue and dequeue) on a sequence encoded therein, the method comprising: executing as part of a pop operation (dequeue operation), an atomic update of a list node and both a deleted node indication (mark the object for deletion) and list-end identifier (new head and tail values) corresponding thereto; the deleted node indication marking the corresponding element for subsequent deletion from the list (delink dequeued object from list) (pg. 550-554).

As to claim 13, PRAKASH teaches executing as part of a delete operation, an atomic update of deleted node indication (pg. 553, "The element is removed, and at some point the DEQ mark is reset.") and at least one list-end identifier corresponding thereto (see algorithm wherein head / tail are set to null).

As to claim 14, PRAKASH teaches responsive to the deleted node indication (marking the object to be dequeued), excising a marked node from the list by atomically updating opposing direction pointers impinging thereon (make head and tail null) and the deleted node indication thereto (via Delink dequeued object from list) (pg. 553, algorithm).

As to claim 15, PRAKASH teaches deleting the marked element from the list (delink dequeued object from list) at least before completion of a same-end push or pop operation (pg. 553).

As to claims 18 and 19, PRAKASH teaches wherein the deleted node indication (marking a node for deletion) is encoded integral with an end-node identifying pointer or dummy node (next object pointer is equal to a nil node) (pg. 550-551).

6. Claims 20-23 are rejected under 35 U.S.C. 102(b) as being anticipated by "Simple, Fast, and Practical Non-Blocking and Blocking Concurrent Queue Algorithms" by Maged M. Michael et al. (MICHAEL).

As to claim 20 and 22, MICHAEL teaches a computer program product encoded in at least one computer readable medium (concurrent non-blocking queue data structure and algorithm) comprising: at least one functional sequence providing non-blocking access on a concurrent shared object instantiably as a linked-list (linked list) delimited by a pair of end-identifiers (head and tail pointers); wherein instances of the at least one functional sequence (enqueue and dequeue operations) concurrently executable by plural processors (processors) of a multiprocessor and each include an atomic operation to atomically update one of the end identifiers (head / tail pointers) and a node of the linked-list (insert a node / remove a node) corresponding thereto, wherein for opposing end instances, the atomic updates are disjoint (via compare and swap

operation in operations) for at least all non-empty states of the concurrent shared object (pg. 269-272; pg. 273-274).

As to claim 21, MICHAEL teaches the function sequence includes both push (enqueue) and pop (dequeue) functional sequences (pg. 270).

As to claim 23 reference is made to an apparatus which corresponds to the program product of claim 20 and is therefore met by the rejection of claim 20 above.

7. Claims 1, 2, 6, and 7 are rejected under 35 U.S.C. 102(e) as being anticipated by "Non-Blocking Synchronization and System Design" by GREENWALD.

As to claim 1, GREENWALD teaches a concurrent shared object representation (deque) comprising: a computer readable encoding for a sequence of zero or more values; and access operations (pushTop / popTop / pushBottom / popBottom) defined for access to each of opposing ends of the sequence (deque), wherein execution of any one of the access operations is non-blocking with respect to any other execution of the access operations throughout a complete range of valid states, including one or more boundary condition states (queue is empty or full) (operations either perform or generate an error when the operation determines that the state is empty or full), and wherein, at least for those of the valid states other than the one or more boundary condition states, opposing-end ones of the access operations are disjoint (pg. 217-221). It is inherent in the teachings of GREENWALD that the operation is executed however,



in some instances the operation results in the intended result, i.e. the node is pushed or popped, or the unintended result, i.e. an error is generated.

As to claim 2, GREENWALD teaches the computer readable encoding includes an array of elements for representing the sequence; and wherein the one or more boundary condition states include one or more empty states (empty queue) (pg 217-221).

As to claim 6, GREENWALD teaches the access operations include push and pop operations, including opposing end variants of each (pg. 217-221).

As to claim 7, GREENWALD teaches the access operations include push and pop operations, including opposing end variants of at least one of the push and pop operations (pg. 217-221).

### ***Claim Rejections - 35 USC § 103***

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

9. Claims 9 and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Non-Blocking Algorithms for Concurrent Data Structures" by PRAKASH in view of "A

Art Unit: 2126

Non-blocking Algorithm for Shared Queues Using Compare-and-Swap" by PRAKASH herein PRAKASH2.

As to claim 9, PRAKASH teaches the access operations include push, pop, and delete operations (enqueue / dequeue / interior delete) (pgs. 5-6). However, PRAKASH does not teach the pop operation includes an atomic update of a list node and both a delete node indication.

PRAKASH2 teaches a concurrent shared object wherein concurrent access (enqueue and dequeue) is mediated, at least in part, by performing during execution of each of the pop operations (dequeue operation), an atomic update (via compare and swap double) of a list node and both a deleted node indication (marking the object for deletion) and list-end identifier (new head and tail values) corresponding thereto (pg. 550-554). However, PRAKASH2 does not mention that the list is a doubly linked list. Official Notice is taken in that it is well known in the art at the time of the invention that a doubly linked list is a version of a singly linked list that stores an extra variable to the previous node. It would be obvious that the teachings of PRAKASH2 is equally applicable to a doubly linked list that allows for nodes to be popped from the front of the queue and pushed on the back of the queue. Therefore, it would be obvious to one of ordinary skill in the art to combine the teachings of PRAKASH with PRAKASH2 in order to perform the non-blocking cooperative processing of shared object operations (pg. 558, conclusion).

As to claim 10, PRAKASH teaches the concurrent access is further mediated, at least in part, by performing, during execution of each of the delete operations, an atomic update (via compare and swap operation to change any variable) (pg. 17) of a deleted node indication (node is nil) and at least one list-end identifier corresponding thereto (new head / move head pointer / changing the pointer of the previous node to a subsequent node of the deleted node) (pg. 17; pgs. 10-11; pgs. 6-8).

10. Claim 17 is rejected under 35 U.S.C. 103(a) as being unpatentable over "A Non-blocking Algorithm for Shared Queues Using Compare and Swap" by PRAKASH.

As to claim 17, PRAKASH teaches the atomic update includes execution of a DCAS operation (pg. 553). However, PRAKASH does not mention that the list is a doubly linked list. Official Notice is taken in that it is well known in the art at the time of the invention that a doubly linked list is a version of a singly linked list that stores an extra variable to the previous node. It would be obvious that the teachings of PRAKASH is equally applicable to a doubly linked list that allows for nodes to be popped from the front of the queue and pushed on the back of the queue. Therefore, it would be obvious that the atomic operations of marking a node for deletion in a pop operation are performed on a doubly linked list.

11. Claim 8 is rejected under 35 U.S.C. 103(a) as being unpatentable over "Non-Blocking Synchronization and System Design" by GREENWALD.

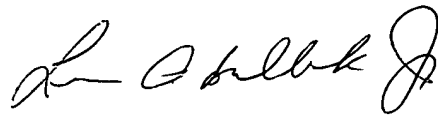
As to claim 8, GREENWALD teaches the array has opposing-end indices respectively identifying opposing ends of the sequence (top / bottom); and wherein concurrent non-blocking access is mediated, at least in part, by performing, during execution of each of the access operations, an atomic update of a respective one of the opposing-end indices and of an array element corresponding thereto (via DCAS operation) (pg. 217-221). However, GREENWALD does not teach that the array is a circular buffer of fixed size. Official Notice is taken in that it is well known in the art that an array can be implemented as a circular buffer of fixed size and therefore would be obvious to one skilled in the art at the time of the invention to implement the array as a circular buffer of fixed size in order to more distinctly identify when the array is full, i.e. front is equal to back.

### ***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lewis A. Bullock, Jr. whose telephone number is (703) 305-0439. The examiner can normally be reached on Monday-Friday, 8:30 am - 5:00 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng An can be reached on (703) 305-9678. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

A handwritten signature in black ink, appearing to read "L. O. Webb Jr.", is located in the lower right quadrant of the page.

lab